

Amendments to the Drawings:

The Examiner rejected Fig. 1 because Fig. 1 should be “designated by a legend such as -- Prior Art -- because only that which is old is illustrated.” As such, the appropriate legend has been added to the replacement figure that accompanies this amendment and response.

The Examiner rejected Fig. 8 because Fig. 8 should be “designated by a legend such as-- Prior Art-- because only that which is old is illustrated.” Applicant respectfully asserts that the figure does not represent “prior art.” Rather, Fig. 8 represents an exemplary implementation of the invention. Fig. 8 has been amended to include components of the present invention in the exemplary embodiment of the computer system as described in the specification. In particular, the parenthetical “(Compiler, Constant Return Optimizer, or Control Operation Optimizer)” was added to the Operating System block 35. The amendment indicates, that in one embodiment, the compiler, constant return optimizer, and/or control operation optimizer is incorporated into the operating system. In addition, the parenthetical “(Constant Tables, vtables, Program Code)” was added to the Program Data block 38. The amendment indicates, that in one embodiment, the constant tables, vtables, and program code is incorporated into the program data. These changes obviate the objection and clearly show that Fig. 8 is not prior art. The changes to the specification and drawings do not add new matter but change the description and drawings to a better form.

The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description: 20 (“computer 20” – page 22 line 14 of the originally filed specification). Fig. 8 has been amended to include the reference sign. Therefore, this objection is also moot.

REMARKS/ARGUMENTS

This Amendment and the following remarks are intended to fully respond to the Final Office Action dated July 26, 2005. In that Office Action, claims 1-23 were examined, and all claims were rejected. More specifically, claims 22 and 23 stand rejected under 35 U.S.C. § 112, second paragraph, as being incomplete for omitting essential structural cooperative relationships of elements, such omission amounting to a gap between the necessary structural connections; and claims 1-23 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over prior art of record “Compiler Transformations for High-Performance Computing” by Bacon et al (hereinafter “Bacon”) in view “Developing a Tool for Memoizing Functions in C++” by McNamee et al. (hereinafter “McNamee”) and further in view of “Improving the Performance of AI Software Payoffs and Pitfalls in Using Automatic Memoization” by Hall et al. (hereinafter “Hall”). Reconsideration of these rejections, as they might apply to the original and amended claims in view of these remarks, is respectfully requested.

In this Response, claims 22 and 23 have been amended; no claims have been canceled; and no new claims have been added.

Claim Rejections – 35 U.S.C. § 112

Claims 22 and 23 stand rejected under 35 U.S.C. § 112, second paragraph, as being incomplete for omitting essential structural cooperative relationships of elements, such omission amounting to a gap between the necessary structural connections. Claim 22 and claim 23 have been amended to put the claims in a more understandable format. As such, this rejection is now moot.

Claim Rejections - 35 U.S.C. § 103

Claims 1-23 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over prior art of record “Compiler Transformations for High-Performance Computing” by Bacon et al in view “Developing a Tool for Memoizing Functions in C++” by McNamee et al. and further in view of “Improving the Performance of AI Software Payoffs and Pitfalls in Using Automatic Memoization” by Hall et al. Applicants respectfully traverse the section 103 rejections. The Examiner has failed to substantiate a prima facie case of obviousness because one or more of the requirements of a prima facie case is absent. Indeed, such a prima facie case can only be met

when **all** of the following requirements are met: (1) there must be some suggestion or motivation in the references themselves (or in the knowledge available to those skilled in the art) to combine the references; (2) there must be a reasonable expectation of success; and (3) the combined references must teach or suggest all the claim limitations. See, MPEP §§ 706.02(j) and 2143. In this case, Bacon, Hall, and McNamee do not describe all of the claim limitations of independent claims 1, 9, and 16. Specifically, Bacon, Hall, and McNamee describe neither generating a return constant table before invoking a target method nor generating an optimized instruction before invoking a target method.

The claims relate to a method or apparatus for optimizing indirect method invocation at a call site. The call site is associated with a receiver object, and programmed to call a target method of a plurality of possible target methods that return constant values. The present invention obviates the need for a function call at these sites. The invention obviates the need for the function call by generating a return constant table, which has constants as table entries. The constants are returned based on specific calls. Before invoking a function or target method, instructions that would normally call the function that generates the constant are replaced with an optimized instruction that merely retrieves the constant value. As a result of this generated “return constant table” and the “optimized instruction,” there is no need to call or execute the function.

Bacon relates to a system or method that constructs a cache to store recent invocation results. Examiner has asserted that Bacon teaches generating optimized instructions at compile time and using those optimized instructions in place of the procedure call. Applicants agree that Bacon teaches generating optimized instructions, but Bacon simply does not teach generating the optimized instruction before invoking the target function. In fact, Bacon explicitly states that memoization caches, “the results of recent invocations.” Bacon, Section 6.8.9. While Applicants do not necessarily agree that Bacon teaches generating optimized instructions at compile time because the cache, for which the optimized instruction calls, has yet to be created at compile time, Bacon does require invoking the target function. The present invention improves on the methods used in past, as embodied by Bacon, because the present invention, as defined in the claims, does not invoke the target method to generate either the return constant table or the optimized instruction.

Applicants do not agree that Bacon does not teach away from generating the return constant table before execution of the target function. Regardless, the combined references must teach or suggest all the claim limitations. See, MPEP §§ 706.02(j) and 2143. As Examiner admits, Bacon does not teach, “population of the constant table prior to invoking the target method.” Office Action, Section 5.

McNamee does not satisfy the inadequacies of Bacon. McNamee describes the development of an automated memoization utility. See, page 17, column 2, paragraph 1. McNamee also does not create, before invoking a target method, an optimized instruction, which obviates the need for a function call, or a return constant table. Instead, McNamee specifically states that a, “new function should check to see if it has been called before with the same argument, return the previously calculated value if so, and if not, invoke the original function...” Page 17, column 2, paragraph 2. For the same reasons as Bacon, Examiner implies that McNamee, does not teach generating the return constant table before invoking the target function. See, Office Action, Section 7. Thus, by Examiner’s own admission, the combination of Bacon and McNamee do not teach generating the return constant table before invoking the target function.

Examiner relies on Hall to teach generating the return constant table before invoking the target function. Hall describes a memoization technique that executes functions offline to minimize the time a function executes during invocations at runtime. Yet, as with Bacon and McNamee, Hall simply does not teach generating the return constant table before invoking the target function. Applicants wish to bring to Examiner’s attention that the portion of Hall that was cited in the Office Action actually requires, “an off-line execution of the expensive routine.” Hall, Section 3.3. While Hall does not memoize the function by invocations at runtime, Hall still executes the function being memoized, i.e., invokes the target method offline. Thus, Hall also does not teach generating the return constant table before invoking the target function.

In general, any prior art that describes memoization will not anticipate or render obvious the claims of the present invention. Memoization, as understood in the art, requires the execution of the function, method, or object to build a cache of results. See, Hall, Section 2; Bacon, Section 6.8.9; McNamee, Section 1. While the execution of the methods may occur at different times, this execution requirement is essential in any memoization. The present invention, as defined in the claims, does not require any such execution of the function or

method but can generate the return constant table and optimized instructions before invoking the target method(s). As such, the present invention eliminates the requirement for execution of the function used in memoization.

The combination of Bacon, McNamee, and Hall simply does not teach or suggest each of the elements of the claimed invention. Bacon, McNamee, and Hall, whether alone or in combination, fail to disclose the generation of an optimized instruction before invoking a target method as recited in claims 1, 12, and 22. Given that these references, both alone and in combination, fail to disclose, teach, or suggest all the claim limitations, they cannot, as a matter of law, render the claims obvious. Reconsideration of the § 103(a) rejections is therefore respectfully requested.

Claims 2-11, 13-21, and 23 depend from these independent claims, and thus, the dependent claims should be allowed for at least the same reasons, namely that the combination of the cited references does not teach the present invention.

Conclusion

This Amendment fully responds to the Office Action mailed on July 26, 2005. Still, that Office Action may contain arguments and rejections and that are not directly addressed by this Amendment due to the fact that they are rendered moot in light of the preceding arguments in favor of patentability. Hence, failure of this Amendment to directly address an argument raised in the Office Action should not be taken as an indication that the Applicant believes the argument has merit. Furthermore, the claims of the present application may include other elements, not discussed in this Amendment, which are not shown, taught, or otherwise suggested by the art of record. Accordingly, the preceding arguments in favor of patentability are advanced without prejudice to other bases of patentability.

It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

Application No. 09/822,535

In light of the above remarks and amendments, it is believed that the application is now in condition for allowance, and such action is respectfully requested. Should any additional issues need to be resolved, the Examiner is requested to telephone the undersigned to attempt to resolve those issues.

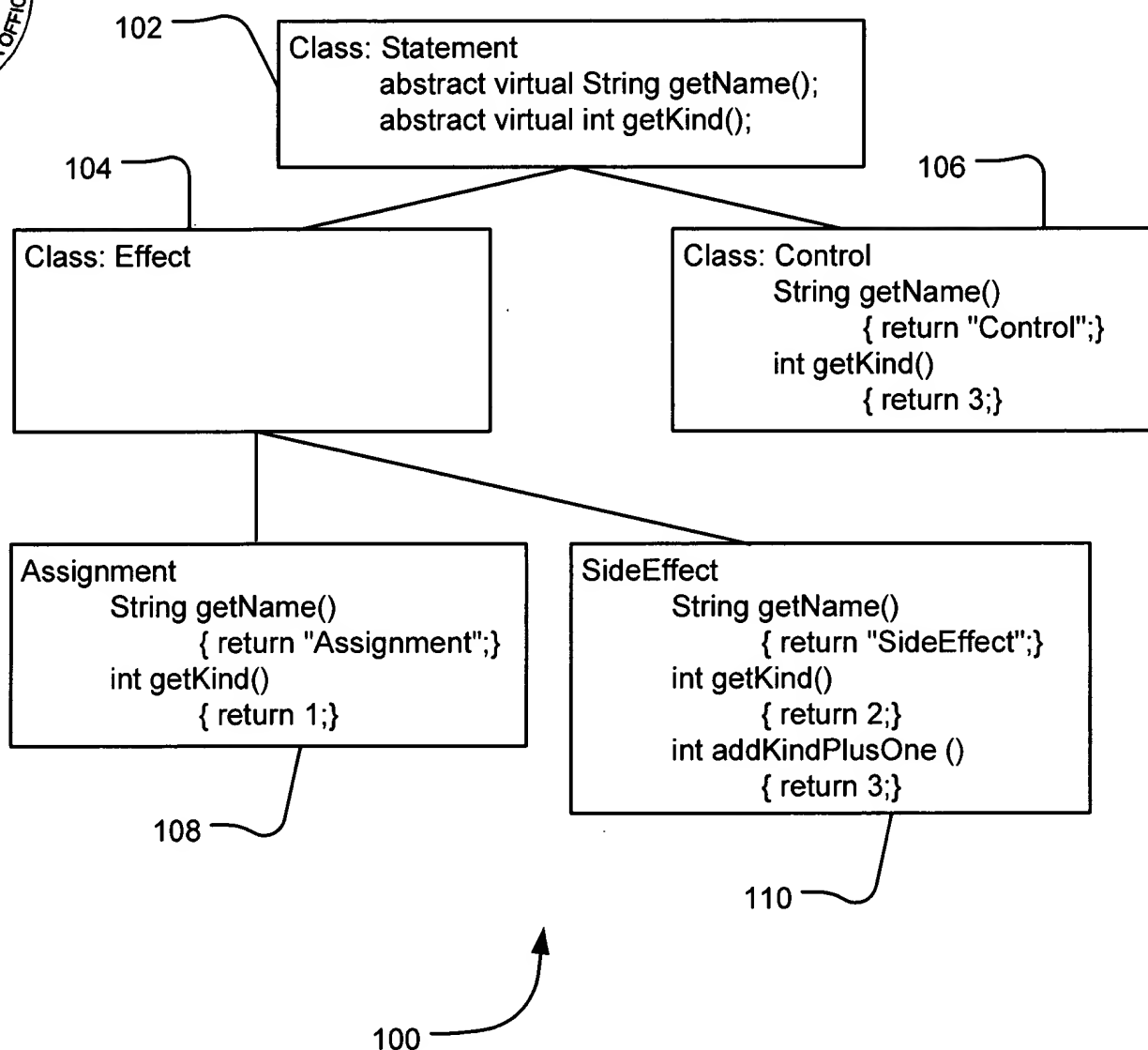
Respectfully submitted,

Dated: 9/19/05



A handwritten signature in dark ink, appearing to read "Tadd F. Wilson". The signature is fluid and cursive, written over a horizontal line.

Tadd F. Wilson, #54,544
MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903
303.357.1651



-- Prior Art--

FIG. 1

FIG. 8

